Deep Learning Project 1
Weight Sharing & Auxiliary Loss

Etienne Salimbeni, Matthias Zeller, Fatih Mutlu
EPFL, Switzerland

***Abstract*—It is crucial to understand the impact of deep learning techniques on simple examples, before diving to complex problems. This project is a brief study on the influence of weight sharing and auxiliary loss on neural networks for a simple binary classification: comparing two digits (greater or equal) from 2 images in the MNIST dataset.**

As a result, while we started from a fully connected neural network base model with an accuracy of 0.80, weight sharing models added a small improvement with an average accuracy of 0.82, whereas auxiliary loss notably improved it to 0.88 and the hybrid models (mix of auxiliary loss and weight sharing) achieved the best performance at 0.91.

I. Introduction

Deep Learning is the foundation of state-of-art models in different tasks such as object recognition, image generation and language translation. The MNIST database is often used as a reference for evaluating new architectures or techniques in image classification. For this study, instead of determining the digit of one MNIST image, we evaluate given a pair of 14x14 grayscale MNIST images if the number on the first is greater than the second.

We will focus on implementing and comparing the performance of a Fully-Connected Neural Network (FCNN) and of a Convolutional Neural Network (CNN) enhanced by Weight Sharing (WS) and Auxiliary Loss (AL) techniques.

II. Models Description

On the following models, we're applying ReLU activation function to the output of each hidden layer and sigmoid to the output layer(s), if not explicitly said otherwise. Along the report, each NN has a unique ID (e.g. FCNN_2, WS_1, AL_1, etc.), to enable easy comparison in the results section.

A. Fully Dense (Base models)

The base model of the study is a FCNN, which is simply a fully connected neural network. We implemented the configurations presented in Fig. 1, testing different number of nodes and layers as well as splitting the network in two independent layers, one per image.

B. Weight Sharing

Weight sharing can be used whenever there is a task-specific network. In our situation, the two input channels represent the same object, i.e. digits. Hence, we can use the same parameters in the digit detection part. The advantage is a reduction in the number of parameters which ultimately leads to faster training [1].

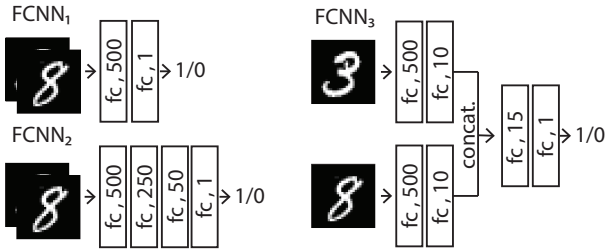

Fig. 1. Illustration of implemented FCNNs. FCNN_1: a FCNN with one hidden layer. FCNN_2: a FCNN with 3 hidden layer. FCNN_3: FCNN for each image individually, then concatenated.

Convolution layers also make use of WS because a specific convolution layer's filter weights are identical over the entire image, which assumes that that if a feature is detected at some position, then the same filter will detect this feature at other positions.

A different way to enable weight sharing is to share the layers themselves, i.e. pass both images through the same layers.

The two weight sharing paradigms are illustrated in Fig. 2.

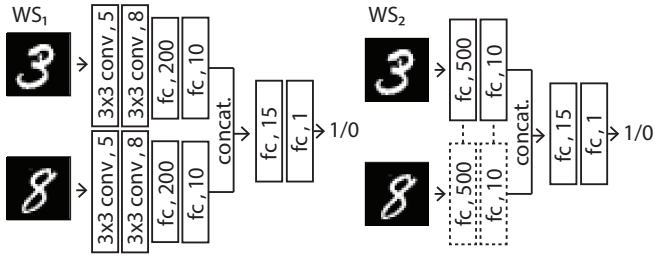

Fig. 2. Structure of different WS models. WS_1: CNN is performed on each image separately. WS_2: a FCNN where the image specific layers are shared (dotted lines).

We also tested a mix of both models, a model including CNN with shared layers (defined as WS_3).

C. Auxiliary Loss

The idea behind auxiliary loss is to discriminate some features at intermediate layers [2]. In this way, the back-propagation signal contains more information and provides regularization. The auxiliary loss will thereby be added to the total loss : $\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{main}} + \lambda \cdot \mathcal{L}_{\text{aux}}$, where λ is arbitrarily chosen.

Fig 3 depicts two flavors of models with auxiliary loss. AL_1 is based on a FCNN, whereas AL_2 is based on a CNN. Fig 4

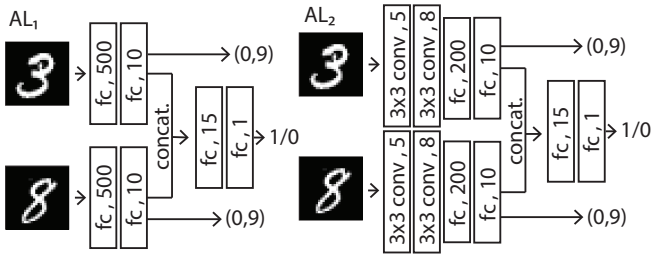

Fig. 3. Architecture of AL models. AL_1: $FCNN_3$ model with auxiliary loss. AL_2: WS_1 model with auxiliary loss.

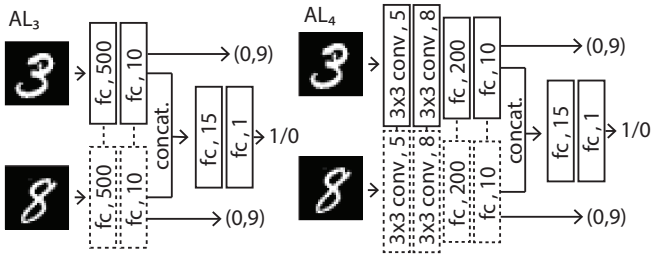

Fig. 4. Architecture of Hybrid (AL+WS) models. AL_3: AL_1 model with WS. AL_4: AL_2 with WS.

displays two hybrid models, i.e. models with weight sharing enhanced with an auxiliary loss.

We tested the model presented in Fig.3. We also tested WS_2 and WS_3 enhanced with auxiliary loss. We'll relate to these models respectively as AL_3 and AL_4 (see Fig.4). Note that each model using AL has its counterpart not using AL : $(AL_1 \leftrightarrow FCNN_3)$, $(AL_2 \leftrightarrow WS_1)$, $(AL_3 \leftrightarrow WS_2)$, $(AL_4 \leftrightarrow WS_3)$.

III. Results

We are using a batch size of 20 for all our tests. Using different batch sizes may yield different results. We chose 20 with a brute force approach, considering the trade-off between speed and accuracy.

Furthermore, all the following training use Adam optimiser. The learning rate chosen for each model is model-specific. As the training time is small, we simply ran brute force search to find the learning rate (which are reported in appendix A).

Lastly, we used the BinaryCrossEntropy loss for the main loss. A sigmoid is applied to the output of the last layer and the digit recognition in the auxiliary task is done with CrossEntropy loss. The λ factor of the AL loss is set to 1 as both losses have similar scale. However, this can be tuned if one wants to give more importance to one loss with respect to another.

The results shown are statistics over 10 runs for each model. Fig. 5 compares the accuracies of the FCNN models when varying the number of layers and neurons per layer, Fig. 6 compares weight sharing models with FCNN, Fig. 7 compares for auxiliary loss models with other models.

Table I recapitulates the accuracies of the main models.

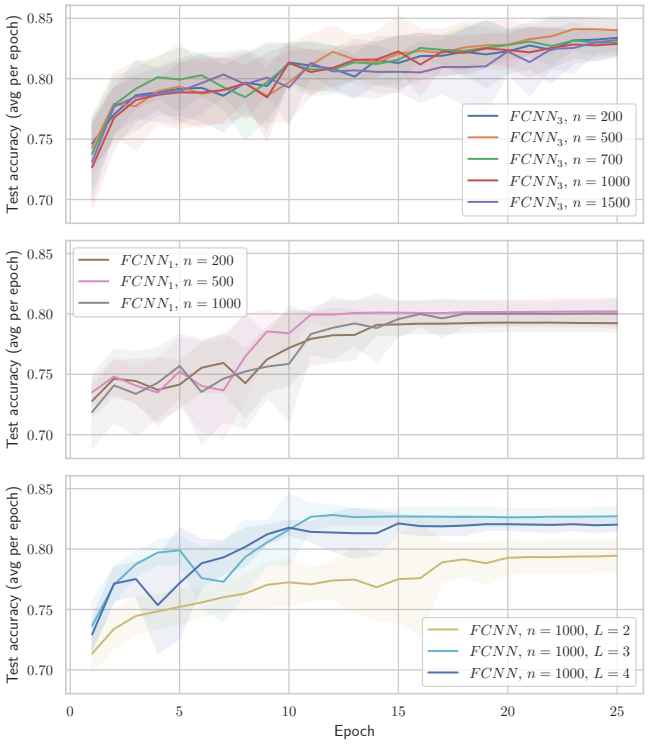

Fig. 5. This graph makes an accuracy comparison by varying elements of the network structure of FCNN. It shows the mean accuracy and standard deviation of it over 10 run. Different number of layers and nodes are tested as well as splitting the layer into independent layer per image (as described in last section).

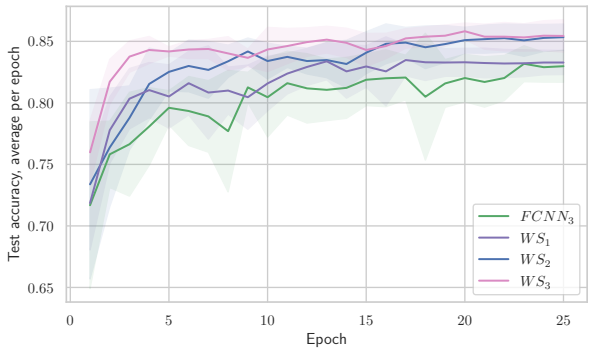

Fig. 6. Comparison of different weight sharing model, as described in last section. The baseline model is $FCNN_3$ which doesn't make use of WS. It shows the mean accuracy and standard deviation of it over 10 run.

IV. Discussion

Before discussing the result, it is important to remember that many factors influence a models performance, and therefore even though we are slightly changing the model to focus the study on AL and WS, other factors like the learning rate or the number of parameters can yield drastically different results.

Fig. 5 shows that simply playing with the number of nodes or layers and splitting the layers per image does not significantly improve the accuracy of the models. The best

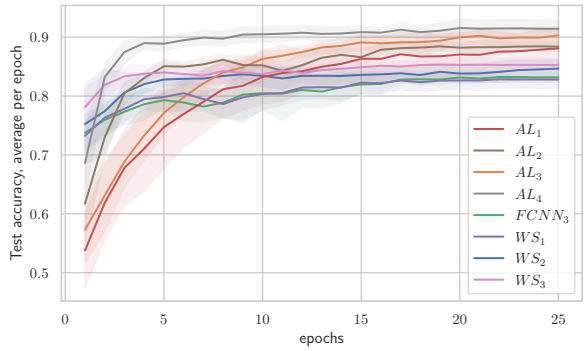

Fig. 7. A graph for comparison of different models using Auxiliary loss. Each model using AL has its equal not using AL : $(AL_1 \rightarrow FCNN_3)$, $(AL_2 \rightarrow WS_1)$, $(AL_3 \rightarrow WS_2)$, $(AL_4 \rightarrow WS_3)$. It shows the mean accuracy and standard deviation of it over 10 run.

TABLE I
PERFORMANCE COMPARISON

		Test Acc. ± Std. Dev.	Model ID
FCNN	Merged Images	0.802 ± 0.011	$FCNN_1$
	Separate Images	0.831 ± 0.008	$FCNN_3$
Weight Sharing	CNN	0.827 ± 0.014	WS_1
	FCNN	0.845 ± 0.014	WS_2
Auxiliary Loss	CNN	0.884 ± 0.007	AL_2
	FCNN	0.881 ± 0.032	AL_1
Aux. Loss and Weight Sharing	CNN	**0.914 ± 0.015**	AL_4
	FCNN	0.902 ± 0.012	AL_3

model is $FCNN_3$ with 0.82 at test accuracy and the worst $FCNN_1$ with 0.80. Similarly, weight sharing did not significantly increase in accuracy but enables faster convergence (see Fig. 6). However, adding the auxiliary loss indeed drastically improves the prediction performance (see Fig. 7).

Our assumption is that, because we simply use convolution layer without applying padding that would keep the dimensionality, and are already using very small images, the CNN approach do not yield a huge improvement compared to the simple FCNN. On the other hand, the AL approach provides stronger regularization, because it takes advantage of an other criteria, namely guessing the digit, which helps the primary task.

V. CONCLUSION

In this work, we proposed deep learning architectures using CNN, FCNN with and without WS, and AL on the MNIST database.

To sum up, prior knowledge about the task is fundamental to improve a NN architecture. For our task, AL had a considerable impact on the test accuracy (+0.1), while WS only added +0.03. The best model results from the combination of both methods. Lastly, it is important to stress that this is a task-specific report and that for other projects AL and WS may not bring the same effects.

Future work could involve making use of other classical regularization techniques (e.g. Dropout and Max-Pooling) or more recent ones (e.g. DropBlock) to reduce overfitting.

REFERENCES

[1] N. L. P. B. Jordan Ott, Erik Linstead, "Learning in the machine: To share or not to share?" 2019. [Online]. Available: https://doi.org/10.1016/j.neunet.2020.03.016

[2] Y. J. Christian Szegedy, Wei Liu, "Going deeper with convolutions," 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

APPENDIX

Model ID	LR for Adam
AL_1	0.0005
AL_2	0.0008
AL_3	0.0005
AL_4	0.002
WS_1	0.0008
WS_2	0.0005
WS_3	0.0013
FCNN_1	0.0005
FCNN_2	0.00015
FCNN_3	0.00015

TABLE II
LEARNING RATES USED FOR THE MODELS, AS DETERMINED BY GRID SEARCH